Remarks

This Amendment is in response to the Office Action mailed 16 July 2002 (16.07.2002). In this Office Action, the Examiner objected to the specification due to an improper incorporation by reference of two US patent applications, and additionally, required the legend “Prior Art” to be added to Table 1. The Examiner also required the submission of formal drawings, with the legend “Prior Art” added to FIG. 1. Finally, the Examiner rejected claims 1-25 under 35 USC 102(b) as being anticipated by Giramma (US Pat. No. 5,706,476) and by the IEEE Standard Multivalue Logic System for VHDL Model Interoperability (Std_logic_1164).

For the Examiner's convenience, included with this amendment are the following:

1. A version of the claims with markings to show the changes made in this amendment.
2. A clean copy of the currently pending claims without parenthetical statements.
3. A version of the replacement paragraphs for the Specification with markings to show the changes made in this amendment.
4. New formal drawings.

1. Objection to the Specification

The Examiner objected to the specification due to an improper incorporation by reference of two US patent applications filed on the same date as the instant application. Specifically, the specification failed to include the serial number for US Pat. App. Ser. No. 09/406,017, filed 9-24-99, entitled “Method and Apparatus For a Monitor that Detects and Reports a Status Event to a Database” and the serial number for US

Pat. App. Ser. No. 09/406,016, also filed 9-24-99, entitled "Method and Apparatus that Reports Multiple Microprocessor Events with a Single Monitor." Applicant is amending the specification to correct these references. In addition, Applicant has added the legend "Prior Art" to Table 1 as required by the Examiner, and has made other minor ministerial changes. Applicant is not adding new matter with these amendments. Therefore, Applicant respectfully requests that the Examiner withdraw the objections to the specification.

2. Claim Rejections Under 35 USC 102(b)

The Examiner rejected claims 1-25 under 35 USC 102(b) as being anticipated by Giramma (US Pat. No. 5,706,476), and by the IEEE Standard Multivalue Logic System for VHDL Model Interoperability (Std_logic_1164).

Giramma discloses a methodology by which a binary-based logic design can be simulated using a combination of an 8-state simulation model and a 4-state simulation model. Giramma's 8-state model is a modification of the standard, well-known MVL-9 model discussed in the instant application and in the IEEE Specification Std_logic_1164, and his 4-state model is a modification of the well-known prior art 4-state model discussed in the instant application. Like the discussion in the current application at page 5, line 14 through page 6, line 20, Giramma discusses the impracticality of using extended-state models, such as an 8- or 9-state simulation model, to simulate complex binary gates, because the size of the truth-tables involved cannot be accommodated in the memory of a standard computer simulation workstation. The present application discloses that designers attempting to employ

extended-state simulations for complex binary logic have developed workaround solutions, such as mapping large truth table functions to multiple smaller truth table functions, or ignoring certain states within a multiple state model for certain designs. Application at p. 6, lines 16-20. In fact, Giramma's method is a prime example of the prior art workaround solutions discussed in the present application, because Giramma has combined both approaches to produce a more efficient binary logic simulation methodology than a straight extended-state approach. In a nutshell, Giramma employs an extended 8-state model only on certain gates in a logic path. For all other gates, Giramma maps certain states within the 8-state model to one of the states of the 4-state model, and then simulates the gates' outputs using the modified and mapped 4-state model. Giramma has redefined the standard 4-state model to ignore the high-impedance (Z) state, and has modified the MVL-9 model to ignore the undefined (D) state. Therefore, Giramma's invention both ignores certain states of the standard models, and maps states of the extended-state model to states of a lower-state model in order to use a smaller truth table to determine a gate's output. This is precisely the kind of prior art discussed in the instant application that is only applicable to binary-based logic designs. As discussed in the instant application, using this kind of approach to simulate even a few gates of an N-NARY design is virtually impossible. The present invention is thus an entirely new method and apparatus that enables designers to simulate *nonbinary* logic, such as N-NARY logic. As such, it is very different from prior art methods like Giramma's invention or the standard extended-state models discussed in the IEEE Specification.

Part of the Examiner's confusion is undoubtedly related to terminology. Treatises that discuss prior art multiple-state simulation models for binary logic (including the IEEE standard and including Giramma) often use the phrases "logic state," "logic value", and "decimal value" somewhat interchangeably. For example, Giramma describes the various states in an extended-state simulation abstraction as "strengths or values" at col. 1, lines 39, 42, and 47, and describes the various states in his 8-state abstraction as having "decimal values" in Table 2. However, in Applicant's lexicon as applied to a binary signal, the phrases "logic value" and "decimal value" refer only to the true-or-false, high-or-low, 1-of-0 nature of the signal—the information used to determine whether a binary signal is asserted or not. In Applicant's lexicon, a signal's "logic value" or "decimal value" bears no relation to its drive state or whether the signal has a status other than asserted or not asserted (i.e., whether a signal is undefined, uninitialized, or high-impedance). In the present invention, a signal model is a representation of a signal that is in a state that includes its logic value, its signal strength, and its signal status—all three of which are separate pieces of information about the signal.

Using Applicant's lexicon, then, every prior art simulation model used for binary logic, whether it employs the simplest 2-state logic model or the most complex extended-state model, can only simulate signals that have a "decimal value" or "logic value" of 1 or 0, regardless of the terminology used that may confuse a signal's "logic value" or "decimal value" with the number of states that the model can represent. In the more sophisticated models, as described in the instant application and in Giramma, these binary signals having a decimal value or logic value of either 1 or 0 also have a "drive state," which may be either strongly driven (typically denoted as S) or weakly-

driven (typically denoted as R). Notably, a strongly-driven high-voltage binary signal does not have a different signal value, decimal value, or logic value than a weakly driven high-voltage binary signal; both signals are asserted and have a logic value of 1. They simply have different drive strengths. Similarly, like a signal's drive strength, prior art simulation models can simulate other characteristics of signals that are not the same as the signal's logic value—for instance, a signal with unknown or indeterminate value (X), a high-impedance signal (Z), an uninitialized signal (U), or an undefined signal (D). These kinds of signals have no logic value—they are unknown, high-impedance, uninitialized, or undefined. These are not "logic values," they are signal characteristics that can be meaningfully simulated.

The imprecision regarding the prior use of the phrases "logic value", "signal value" and "decimal value," to describe a modeled signal having a certain signal state that may include a logic value, a drive strength, and a signal status has led to the use of the terms "multivalue logic" and "multiple value logic" to describe models that include extended-state model abstractions, even though these extended-state models are used to simulate binary logic, which is not multivalue logic. Applicant's use of the terms "multiple value logic" or "multivalue logic" means nonbinary logic, i.e., N-NARY logic. In the present invention, and using Applicant's precise terminology, a 1-of-N N-NARY signal's "logic value" or "decimal value" means the value encoded in the signal's multiwire bundle, as indicated by which one of the wires within the signal bundle is asserted. See Table 2, page 7. Therefore, in the present invention, an N-NARY signal model can have more than two logic values—indeed, virtually every signal in the N-NARY logic family has more than two logic values. The signal illustrated in Table 2 has

four logic values or decimal values—this signal can have a logic value or decimal value of 0, 1, 2, or 3. In the present invention, a signal having one of these logic values may also have a certain drive strength—it may be strongly driven, moderately driven, or weakly driven—*but an N-NARY signal's drive strength does not change its logic value*. As in the prior art methodologies, and using Applicant's definition a signal's logic value, drive strength (strong or resistive) is simply a description of a signal characteristic that is not the same as the signal's logic value.

Using Applicant's lexicon, then, neither Giramma nor the IEEE specification disclose the simulation of multivalue signals—they simply disclose the simulation of binary signals, wherein the signal model is capable of simulating the binary signal's logic value—0 or 1—plus a number of characteristics that includes the drive strength of the logic value, or the fact that a binary signal has no logic value or drive strength at all, but rather, is undefined, uninitialized, or high-impedance. While the IEEE reference refers to the undefined, unitialized, and high-impedance signal states as "logic values," in Applicant's parlance, these are more properly considered to be signal characteristics. In Applicant's parlance, a 1-of-N signal's logic value is an integer that carries meaningful encoded information rather than signal status information such as undefined, unitialized, or high-impedance.

Given this clarification of terminology, the difference between the IEEE specification binary-based methodologies, or Giramma's methodology (which is simply a modification and combination of the IEEE methods), and the present invention is clear. The IEEE models and Giramma's method are applicable only to binary logic, and simulate only binary signals and their various characteristics of interest to binary logic

designers. The present invention models nonbinary signals—signals capable of having meaningful logic values, independent of drive strength or other signal characteristics, that are something other than 0 or 1. In addition to the logic value, the present invention models various characteristics of nonbinary, 1-of-N signals used in N-NARY logic that are of interest to N-NARY logic designers, such as drive strength and signal status. As disclosed in the instant application, the present invention enables the simulation of N-NARY logic using the arithmetic or logical manipulation of the signal models, thus freeing the simulation from the need for cumbersome truth tables. Accordingly, the present invention is quite different from the prior art.

Notwithstanding, given the imprecise terminology used in the IEEE specification and other prior art and the resulting confusion, Applicant has amended the claims to clarify and highlight the differences between the present invention and the prior art. Specifically, amendments made to claims 1, 6, 11, 16, and 21 make it very clear that the claimed signal model is a model of a nonbinary 1-of-N logic signal. Neither Giramma nor the IEEE specification Std_logic_1164 disclose or describe an extended-state modeling method or apparatus applicable to nonbinary 1-of-N signals that are unique to N-NARY logic. Consequently, given these amendments, neither Giramma nor the IEEE specification Std_logic_1164 is a proper §102 reference because neither teaches or shows all of the elements of the currently claimed invention.

As described in the instant application and mentioned briefly above, a primary feature of the present invention is the capability the signal model provides to execute simulations of N-NARY logic by mathematically and/or logically manipulating modeled input and output signals, thereby avoiding complex truth tables or the need to break

gates into logical primitives. This is a vast improvement over the prior art methods disclosed by Giramma, using the IEEE models or derivatives thereof. Although, as described above, the addition of the N-NARY signal limitation into the claims removes Giramma and the IEEE specification as proper §102 references, a few of the Examiner's specific contentions regarding claim limitations must also be addressed.

First, as discussed above, Applicant's use of the term "decimal value" or "logic value" means the logic value of the signal. In a binary signal, that logic value can only be 0 or 1, regardless of the drive state or other signal characteristics. In nonbinary N-NARY logic, 1-of-N signals can have N logic values, typically described as integers from 0 to (N-1). See Table 2. Consequently, neither Giramma nor the IEEE specification teach modeling a 1-of-N logic signal having a logic value that ranges from greater than 1 or equal to or less than 31. Giramma's 32-bit "zoom" words have nothing to do with the logic value of a modeled signal; they are simply a convenient methodology to access the truth tables Giramma uses to determine a gate's output and send "directives" (i.e., inputs and state abstractions) to the next downstream gate. As the present invention is used in a simulation that does not utilize truth tables, Giramma's 32-bit "zoom" words are wholly unrelated to any feature or function of the present invention.

Similarly, Giramma's assignment of a "decimal value" to each modeled state (See Table 2, col. 6) is completely arbitrary, the "decimal value" is only a pointer that cross-references to a specific modeled state and does not relate to the actual logic value (0 or 1) or drive state (S or R) or other signal characteristic (U, Z, or X) capable of being modeled. In contrast, the signal value in the claimed signal model is the exact

logic value or decimal value of the 1-of-N signal being modeled. The signal value of the model must be identical to the decimal value or logic value of the signal being modeled or the simulation will not run correctly.

Since neither Giramma nor the IEEE specification teaches the elements and limitations discussed above, the claimed invention is novel over these references. Specifically, neither Giramma nor the IEEE specification teaches a signal model that includes a signal value which is the logic value of a nonbinary 1-of-N logic signal being modeled, wherein the logic value further comprises an integer greater than 1. Therefore, Applicant respectfully requests that the Examiner withdraw the rejection to claims 1-25 under 35 USC 102(b) as being anticipated by Giramma (US Pat. No. 5,706,476), and by the IEEE Standard Multivalue Logic System for VHDL Model Interoperability (Std_logic_1164).

3. Summary

In view of the above, Applicant believes that each of the presently pending claims is in immediate condition for allowance or appeal. Accordingly, Applicant respectfully requests that the Examiner withdraw the outstanding objections and rejections of the claims and pass this application to issue.

Respectfully submitted,

Karen S. Wright

Date: 16 October 2002

Karen S. Wright
Reg. No. 45,240

Matthew J. Booth
Karen S. Wright
Booth & Wright, L.L.P.
PO Box 50010
Austin, Texas 78763-0010
Telephone: (512) 474-8488
Facsimile: (512) 474-7996
matthew.booth@boothlaw.com
karen.wright@boothlaw.com
http://www.boothlaw.com/

Version Of Claims With Markings To Show Changes

CLAIMS

We claim the following invention:

1. (First Amended) A signal model used in an N-NARY logic simulation, ~~that simulates logic signals capable of having more than two unique decimal values and one or more unique drive states,~~ comprising:

a signal value ~~field~~, said signal value ~~field~~ further comprises ~~information that conveys the decimal~~ logic value of ~~the~~ a nonbinary 1-of-N logic signal being modeled, wherein said logic value further comprises an integer greater than 1;

a signal strength ~~field~~, said signal strength ~~field~~ further comprises ~~information that conveys~~ the drive state of ~~the~~ said nonbinary 1-of-N logic signal being modeled; and

a signal definition ~~field~~, said signal definition ~~field~~ further comprises ~~information that conveys whether the signal being modeled holds a defined value or an undefined value.~~ the defined or undefined status of said nonbinary 1-of-N logic signal being modeled.

2. (First Amended) The model of Claim 1, wherein said ~~signal value field is capable of holding up to 32 unique decimal signal values.~~ logic value further comprises an integer less than or equal to 31.

3. (First Amended) The model of Claim 1, wherein said signal strength ~~field is capable of conveying whether said decimal value of the logic signal being modeled is in the,~~ further comprises one of the following: the high-impedance state, the weakly-driven state, the moderately-driven state, or the strongly-driven state.

4. ~~The model of Claim 3, wherein said signal strength field conveys that said decimal value of the logic signal being modeled is strongly driven when said signal strength field is set to "00".~~ Canceled.

5. ~~The model of Claim 1, wherein said signal definition field holds the value of "0" when said logic signal being modeled is a valid, defined signal.~~Canceled.

6. (First Amended) A method that ~~models logic signals capable of having more than two unique decimal values and one or more unique drive states for use in~~ makes a signal model used in an N-NARY logic simulation, comprising:

~~defining~~ assigning a signal value, ~~field that comprises information that conveys the decimal~~ said signal value further comprises the logic value of ~~the~~ a nonbinary 1-of-N logic signal being modeled, wherein said logic value further comprises an integer greater than 1;

~~defining~~ assigning a signal strength ~~field that~~ , said signal strength further comprises ~~information that conveys~~ the drive state of ~~the~~ said nonbinary 1-of-N logic signal being modeled; and

~~defining~~ assigning a signal definition ~~field that comprises information that conveys whether the signal being modeled holds a defined value or an undefined value.~~, said signal definition further comprises the defined or undefined status of said nonbinary 1-of-N logic signal being modeled.

7. (First Amended) The method of Claim 6, wherein said ~~signal value field is capable of holding up to 32 unique decimal signal values.~~logic value further comprises an integer less than or equal to 31.

8. (First Amended) The method of Claim 6, wherein said signal strength ~~field conveys whether said decimal value of the logic signal being modeled is in~~further comprises one of the following: the high-impedance state, the weakly-driven state, the moderately-driven state, or the strongly-driven state.

9. ~~The method of Claim 8, wherein a value of "00" in said signal strength field conveys that said decimal value of the logic signal being modeled is strongly driven.~~ Canceled.

10. ~~The method of Claim 6, wherein said signal definition field holds the value of "0" when said logic signal being modeled is a valid, defined signal.~~ Canceled.

11. (First Amended) A method that uses a~~logic~~ signal model ~~in a software-implemented simulation of a logic design, wherein said logic signal is capable of having more than two unique decimal values and one or more unique drive states,~~used in an N-NARY logic simulation, comprising:

reading a signal value, ~~field that~~ said signal value further comprises ~~information that conveys the decimal~~ the logic value of ~~the~~ a nonbinary 1-of-N logic signal being modeled, wherein said logic value further comprises an integer greater than 1 ;

reading a signal strength~~ field that~~, said signal strength further comprises ~~information that conveys~~ -the drive state of ~~the~~ said nonbinary 1-of-N logic signal being modeled;

reading a signal definition ~~field that~~ said signal definition further comprises ~~information that conveys whether the signal being modeled holds a defined value or an undefined value; and~~ the defined or undefined status of said nonbinary 1-of-N logic signal being modeled; and

providing said ~~decimal value, said drive state, and said definition information of the logic signal being modeled~~signal value, said signal state, and said signal definition to the software-implemented simulation of the N-NARY logic design.

12. (First Amended) The method of Claim 11, wherein said ~~signal value field is capable of holding up to 32 unique decimal signal values.~~logic value further comprises an integer less than or equal to 31.

13. (First Amended) The method of Claim 11, wherein said signal strength ~~field conveys whether said decimal value of the logic signal being modeled is in~~further comprises one of the following: the high-impedance state, the weakly-driven state, the moderately-driven state, or the strongly-driven state.

14. ~~The method of Claim 13, wherein a value of "00" in said signal strength field conveys that said decimal value of the logic signal being modeled is strongly driven.~~ Canceled.

15. ~~The method of Claim 11, wherein said signal definition field holds the value of "0" when said logic signal being modeled is a valid, defined signal.~~ Canceled.

16. (First Amended) A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform a method that uses a ~~logic~~ signal model ~~in a software-implemented simulation of a logic design, wherein said logic signal is capable of having more than two unique decimal values and one or more unique drive state, said method comprises:~~ signal model used in an N-NARY logic simulation, comprising:

reading a signal value, ~~field that~~ said signal value further comprises ~~information that conveys the decimal~~ the logic value of ~~the~~ a nonbinary 1-of-N logic signal being modeled, wherein said logic value further comprises an integer greater than 1 ;

reading a signal strength ~~field that~~, said signal strength further comprises ~~information that conveys~~ the drive state of ~~the~~ said nonbinary 1-of-N logic signal being modeled;

reading a signal definition ~~field that~~ said signal definition further comprises ~~information that conveys whether the signal being modeled holds a defined value or an undefined value; and~~ the defined or undefined status of said nonbinary 1-of-N logic signal being modeled; and

providing said ~~decimal value, said drive state, and said definition information of the logic signal being modeled~~signal value, said signal state, and said signal definition to the software-implemented simulation of the N-NARY logic design.

17. (First Amended) The program storage device of Claim 16, wherein said ~~signal~~ logic value ~~field is capable of holding up to 32 unique decimal signal values.~~ further comprises an integer less than or equal to 31.

18. (First Amended) The program storage device of Claim 16, wherein said signal strength ~~field conveys whether said decimal value of the logic signal being modeled is in~~further comprises one of the following: the high-impedance state, the weakly-driven state, the moderately-driven state, or the strongly-driven state.

19. ~~The program storage device of Claim 18, wherein a value of "00" in said signal strength field conveys that said decimal value of the logic signal being modeled is strongly driven.~~ Canceled.

20. ~~The program storage device of Claim 16, wherein said signal definition field holds the value of "0" when said logic signal being modeled is a valid, defined signal.~~ Canceled.

21. (First Amended) A signal modelling system ~~that models logic signals capable of having more than two unique decimal values and one or more unique drive states for use in~~ used in an N-NARY logic simulation, comprising:

~~encoding the decimal value of the logic signal being modeled in a signal value field;~~ a signal value, said signal value further comprises the logic value of a nonbinary 1-

of-N logic signal being modeled, wherein said logic value further comprises an integer greater than 1;

~~encoding the drive state of the logic signal being modeled in a signal strength field; and~~ a signal strength, said signal strength further comprises the drive state of said nonbinary 1-of-N logic signal being modeled; and

~~encoding whether the signal being modeled holds a defined value or an undefined value in a signal definition field.~~ a signal definition, said signal definition further comprises the defined or undefined status of said nonbinary 1-of-N logic signal being modeled.

22. (First Amended) The system of Claim 21, wherein said ~~signal~~ logic value ~~field is capable of holding up to 32 unique decimal signal values.~~ further comprises an integer less than or equal to 31.

23. (First Amended) The system of Claim 21, wherein the drive ~~state encoded in said signal strength field indicates whether said decimal value of the logic signal being modeled is in~~strength further comprises one of the following: the high-impedance state, the weakly-driven state, the moderately-driven state, or the strongly-driven state.

24. ~~The system of Claim 23, wherein a value of "00" encoded in said signal strength field indicates that said decimal value of the logic signal being modeled is strongly driven.~~ Canceled.

25. ~~The system of Claim 21, wherein a value of "0" in said signal definition field indicates that said logic signal being modeled is a valid, defined signal.~~ Canceled.

Version Of Replacement Paragraphs With Markings To Show Changes Made

Paragraph at p. 2, l. 9-12

N-nary logic is a new dynamic logic design style fully described in a copending patent application, U.S. Pat. App. Ser. No. 09/019355, filed 2-5-98, now U.S. Pat. No. ~~__________,~~6,066,965, and titled "Method and Apparatus for a N-Nary logic Circuit Using 1-of-4 ~~Encoding",~~Signals", which is incorporated herein for all purposes and is hereinafter referred to as "The N-nary Patent."

Paragraph at p. 5, lines 3-13.

Table 1 provides a summary of the logic states supported by existing simulation tools currently used to evaluate and verify typical binary-based logic designs.

Table 1-Prior Art

Model state level	Logic states supported	Definitions
2-state	0, 1	
3-state	0, 1, X	X=uninitialized
4-state	0, 1, X, Z	Z="undriven" or high-impedance
9-state	0S, 1S, XS, 0R, 1R, XR, 0Z, 1Z, XZ	S="Strongly" driven; R="weakly" driven; Z=not driven, value of wire=last known value
MVL-9	0S, 1S, XS, 0R, 1R, XR, Z, UI, UD	Z=not driven; last value of wire ignored UI=uninitialized; last value of wire ignored UD=undefined; last value of wire ignored
12-state	0S, 1S, XS, 0R, 1R, XR, 0Z, 1Z, XZ, 0I, 1I, XI	I=Indeterminate

Paragraph at p. 15, l. 8-23:

The simulation environment files 130 can be written in any programming language, but in a preferred embodiment, are written in a version of ANSI C++ that is compatible with the g++ compiler 124 within the Design Tool Compiler 120. The simulation environment files provide the interface between the operator and the simulated logic, and enable the operator to specify test-related variables such as the types of tests to be run in the simulation, input signals, output points, and output content and format. In a preferred embodiment, the simulation environment files will include the monitor and monitoring methods disclosed in the following copending U.S. Patent Applications:

U.S. Pat. App Ser. No.	Date Filed	~~U.S. Pat. No.~~	Title
09/406,017	9/24/99	~~__________~~	Method and Apparatus For a Monitor that Detects and Reports a Status Event to a Database
09/406,016	9/24/99	~~__________~~	Method and Apparatus that Reports Multiple Microprocessor Events with a Single Monitor